````
# LoRA Fine-Tuning Demo on Hugging Face (CPU-Only, Windows-Compatible)

This repository demonstrates **parameter-efficient fine-tuning (LoRA)**
applied to a small Hugging Face transformer model (`distilgpt2`) using the
**PEFT** library. The example is designed to run **entirely on CPU** under
Windows, making it suitable for environments without GPU acceleration.

Only a small fraction of model parameters (≈0.18%) are trained, while the
pretrained base model remains frozen. This setup reflects modern best
practices for adapting large language models under limited computational
resources.

---

## Key Features

- Uses a real pretrained model (`distilgpt2`, ~82M parameters)
- Applies LoRA adapters to attention projections
- Trains only LoRA parameters (parameter-efficient fine-tuning)
- Runs on CPU (no CUDA / GPU required)
- Saves and reloads LoRA adapters separately from the base model
- Fully compatible with Windows + Python 3.12

---

## Environment Setup

Tested environment:

```bash
pip install pyarrow==15.0.2
pip install datasets==2.18.0
pip install transformers==4.38.2
pip install peft==0.10.0
pip install accelerate==0.27.2
````

---

# How It Works

1. Load pretrained `distilgpt2` from Hugging Face.
2. Inject LoRA adapters using the PEFT library.
3. Freeze base model weights; train only LoRA parameters.
4. Save LoRA adapters to disk.
5. Reload base model + adapter for inference.
6. Perform text generation using the adapted model.

This approach reduces memory footprint and training cost compared to full fine-tuning.

---

# Why LoRA?
````

LoRA replaces full weight updates with low-rank updates to selected projection matrices in attention layers:

$$W' = W + \Delta W, \quad \Delta W = AB$$

where $A \in R^{d \times r}$, $B \in R^{r \times k}$, and $r \ll \min(d,k)$

This significantly reduces the number of trainable parameters while preserving adaptation capacity.

---

## Example Output

After training, the model can generate adapted text such as:

```
Today I feel like I deserve to be a part of this game, so I wanted to add a
little flair...
```

---

## Intended Use

- Educational demonstration of LoRA / PEFT
- CPU-only experimentation
- Reproducible minimal fine-tuning example
- Reference implementation for low-resource environments

---

## License

MIT License